1 Objectives

LL parsing is easy to use, but unless the grammar is *just right*, you may need to perform all kinds of tedious[1] transformations. For this reason, it is much more common to use LR parsing.

- Be able to explain how LR parsing is different than LL parsing.
- Know how to generate the Characteristic Finite State Automata for a grammar.
- Know what the item sets mean.
- Know how to create the LR Action and Goto tables
- Know what a Shift-reduce conflict is, and how to fix it
- Know what a reduce-reduce conflict is, and how to fix it
- Know how to use the parse tables to generate an "execution trace" of a parse.

2 Problems

The following problems will help with studying for the exam. We may do some of them in class.

1. Generate the Characteristic Finite State Machine for the following grammar.
   $S \rightarrow b\ A\ e$
   $A \rightarrow x\ E$
   $A \rightarrow x\ I$
   $E \rightarrow a\ N$
   $I \rightarrow c\ N$
   $N \rightarrow y$
2. What is a shift-reduce conflict? How does a parser typically deal with it? Why?
3. Give an example of a grammar that will have a shift-reduce conflict.
4. Give two separate ways of proving that a grammar is unambiguous.
5. Give two separate ways of proving that a grammar is ambiguous.
6. How does a context-free grammar relate to a regular grammar, as far as expressive power? Justify your answer.
7. What is the advantage of an LR parser over an LL parser?

[1] Well, you won't do it yourself, you'll make the computer do it for you.